Journal of Intelligent & Fuzzy Systems xx (20xx) x–xx
DOI:10.3233/IFS-141199
IOS Press

Deliberative control components for eldercare robot team cooperation

José M. Gascueña[a,*], Francisco J. Garijo[c], Antonio Fernández-Caballero[a,b], Marie-Pierre Gleizes[c] and André Machonin[c]
[a]*Instituto de Investigación en Informática de Albacete (I3A), Albacete, Spain*
[b]*Departamento de Sistemas Informáticos, Universidad de Castilla-La Mancha, Albacete, Spain*
[c]*IRIT - Institut de Recherche en Informatique de Toulouse, Toulouse Cedex 4, France*

Abstract. One important research area in autonomous mobile robotics is to create companions that live in our ambience and perform tasks to help in everyday life. On the other hand, Ambient Intelligence (AmI) seeks to create a network of helpful intelligent devices. This paper describes an approach for the development of cooperation models for eldercare robot teams using goal-driven control components. The framework and the approach are illustrated through the development and assessment of task allocation in multi-robot teams. Two cooperation models are implemented: (i) a team model based on the adaptive multi-agent systems theory where task responsibility is agreed among team peers by exchanging individual estimations of the degree of difficulty and priority to achieve the task; (ii) a hierarchical model where a robot manager asks for the estimations of its team members and then assigns the task. Experimentation for team cooperation assessment is performed through considering environmental changes, as well as communications and internal failures. The proposal is simulated in an AmI-oriented elderly care center to assist seniors in need.

Keywords: Adaptive multi-agent systems (AMAS), agent frameworks, robotics, distributed task allocation, cooperation models, ambient Intelligence

1. Introduction

The technological development in robots, computing and communications has led to envisage the design of robotic systems consisting of networked vehicles, sensors, actuators and communication devices [24]. The existing robots are generally grouped into three types such as industrial robots, service robots and robots with special missions [4]. The robots that perform works and service activities directly for human beings are called service robots [25]. Recently, service robots are getting increased attention because of their potential applications for enhancing human well-being and quality of life in the so-called Ambient Intelligent (AmI) paradigm [1]. The convergence of AmI intelligence and autonomous robotics has given birth to several new research areas, including network robot systems, ubiquitous robotics, and robot ecologies [7].

The main goal in all these areas is to design intelligent robotic environments, that is, environments where close communication is established among sensing and robotic devices. The coordinated cooperation of such devices enacts and supports complex tasks to help the users in everyday life. Modern societies face the problem of growing increasingly older, meaning that more effort has to be put into the care of an ever-growing older society. Besides caring for our elderly by ourselves, assistance systems for everyday tasks will become more

*Corresponding author. José M. Gascueña, Instituto de Investigación en Informática de Albacete (I3A), 02071 Albacete, Spain. E-mails: JManuel.Gascuena@gmail.com (José M. Gascueña); fgarijo@gmail.com (Francisco J. Garijo); Antonio.Fdez@uclm.es (Antonio Fernández-Caballero); gleizes,@irit.fr (Marie-Pierre Gleizes); Andre.Machonin@irit.fr (André Machonin).

and more important [9, 27, 28]. So, novel developments enable engineers to design new robotic systems that interact with humans and other robots in a cooperative way [8, 13, 26].

Now, component-based approaches are increasingly used to deal with heterogeneity and complexity of robotic systems [5, 11, 20]. A key advantage of componentization is to allow the development of simulated models which could be seamlessly deployed, fully or in part, into the robot hardware. Ongoing work on robot simulation tools is also in this direction [12]. This is why this paper introduces a component-based layered architecture for mobile eldercare robots which control is based on a deliberative goal-driven agent pattern [22]. High-level deliberative control facilitates development and experimentation with different behavior models by bridging the gap between analysis, design and implementation. It also allows reusability and ease traceability of the control process which is based on high level constructs close to human behavior. However, common pitfalls are hard integration with software engineering standards, poor performance, and difficulty to control the deliberative process. Therefore, integration of symbolic deliberative with imperative components is still a challenge.

Moreover, this paper describes an architectural framework for implementing teams of mobile eldercare robots capable to achieve individual and collective mission goals by taking into account unexpected changes in the environment, internal failure and availability of mission resources. Our work focuses on sensor and data, and the intelligence is embedded outside the devices, which implies a notable delegation for a tier of computing services. While most of the experimental results focus on simulated coordination for *best cases* [10, 28, 29], the most significant results reported in this work concern team coordination in stressing situations. Performance testing has been done considering different team size, tasks to be achieved, and eldercare robot deployment in different processing nodes in order to assess the impact of communication.

The rest of the paper is organized as follows. Section 2 outlines the architectural principles for mobile eldercare robot design, and the rationale for adopting a goal-oriented approach for implementing robot control and team cooperation. This approach is illustrated with the development of two cooperation models in the experimental setting: (i) an adaptive multi-agent system (AMAS) model where each team member evaluates the cost to achieve the goal, sends its evaluation to its peers and then assumes the goal if it has the most suitable evaluation; (ii) a hierarchical model where a manager asks each peer to estimate the evaluation of a given goal, then it proceeds to assign the goal to the most suitable peer. Section 3 details assessment metrics and testing results after using different configurations made up of various team sizes and number of assisted elderlies. Stress testing has been performed to compare both functional and performance issues on AMAS and hierarchical models. Finally, conclusions and open issues are summarized in Section 4.

2. A goal-oriented approach for robot control and cooperation

Successful integration of high-level deliberative decision and control components into mobile robot systems relies on the manageability, autonomy, information elaboration and abstraction of the functional units dealing with sensing, navigation, actuation and, communication capabilities. The proposed approach relies on a multi-layered component-based architecture which is populated by manageable components offering their services to other components through standard interfaces (see Fig. 1). The vertical layer contains information models shared by horizontal layers: sensorial, mediation and control layer.

The sensorial layer gathers the components encapsulating sensory functions such as low level image processing, temperature acquisition, distance evaluation, obstacle detection, energy management, vision, and motion. The mediation layer contains components that process low level information coming from the lower layer to elaborate semantic information, which simplifies tasks and decisions performed at control layer. The perception component aims to process, filter, select and correlate incoming information emitted by the components of the sensorial layer, as well as information received via messages sent by other agents. The persistence component provides persistence services to the upper layer. Actuation and communication components aim to provide high-level services such as moving, message sending and other actions to the components of the control layer.

2.1. The deliberative control component

The Robot Global Control (RGC) in the control layer is in charge of orchestrating the internal component behaviors to achieve a coherent global behavior. The RGC gathers elaborated information from the rest of the components, makes choices, orders execution of

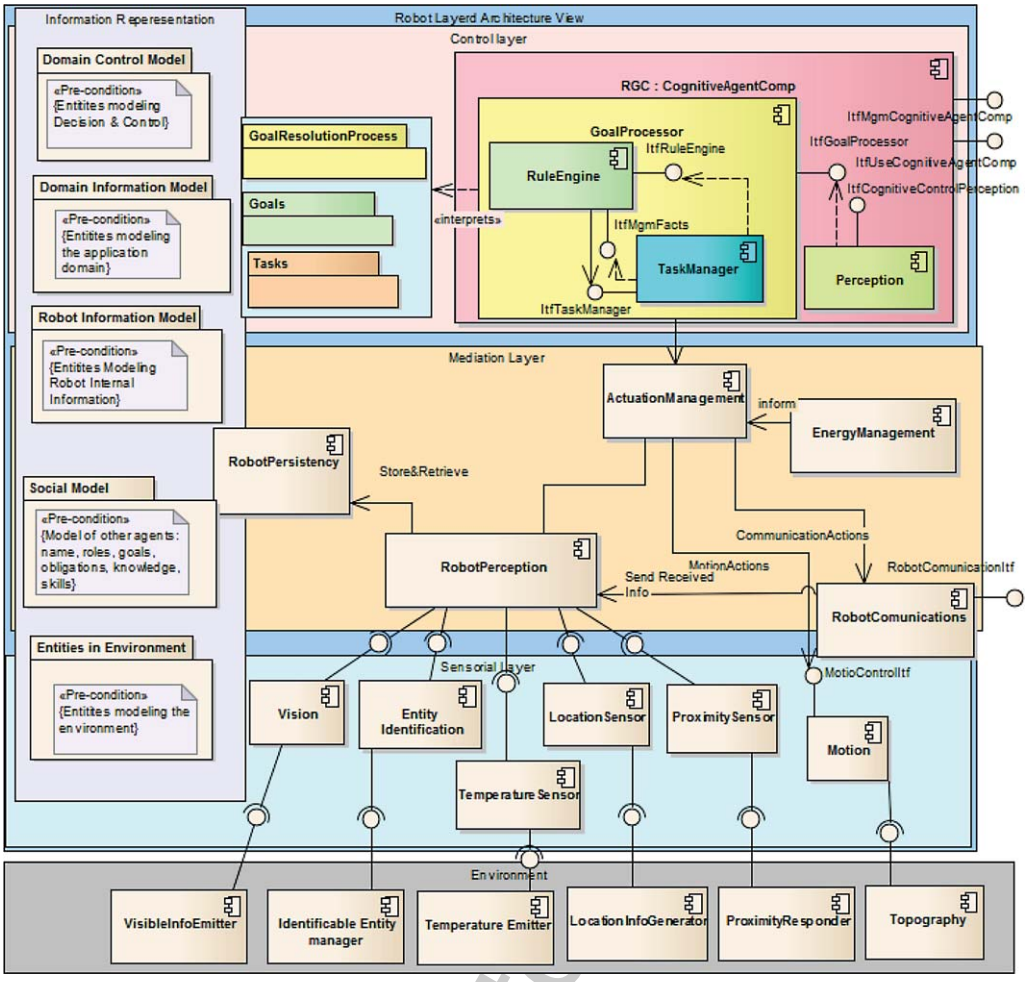

Fig. 1. The general multi-layered component-based architecture.

actions, monitors results, and sends control information to relevant components when necessary.

The RGC control component is implemented with a declarative goal processor that manages a goal space and a working memory [14]. In order to achieve goals, strategic and tactic criteria for generating goals and executing tasks and actions are defined by means of situation-action rules. The situation part specifies a partial state of the working memory including the objective and its internal state, and the action part contains statements for executing tasks. The processing cycle is driven by incoming information which is stored in the working memory. Then, control rules are used to decide either to generate new goals, focus on a new goal, verify the resolution of pending goals, or proceed to the resolution of pending goals by executing new tasks and actions. While the processing model is in line with other deliberative architectures (e.g. [3, 6, 30]), from an engineering perspective there are significant differences: (i) key internal components such as the rule engine and the task manager are implemented with existing open source software (e.g. Drools [2]) and, (ii) multiple behavior models are supported. Multiple concurrent distributed instances are generated from each behavior model. Componentization allows seamless integration of real or simulated components, thus facilitating modeling, encapsulation and reuse of control strategies and cooperation models.

2.2. *Developing team cooperation models in the experimental framework*

Our work on team cooperation focuses on evaluating different control architectures and cooperation

models allowing a robot team to efficiently achieve mission goals. The experimental setting for eldercare robot operation is based on crisis management scenarios. The mobile eldercare robot team is situated in the intervention area to help the elderly in need. The Control Center (CC) broadcast requests to help impaired seniors through indicating priority, location, and additional details when needed.

The team is capable of interpreting and evaluating the CC requests taking into account their current workload, then deciding which member of the team would assume the goal for helping the elderly. Finally, the team mates that have accepted the responsibility of accepting the goal proceed to assist the senior. Indeed, some initial experiments have started implementing the AMAS cooperation model so far where robots are considered AMAS agents with same capabilities (e.g. [17, 18, 21]). Team members have neither information nor explicit representation of the team's objectives. These global objectives are achieved through cooperation among team members. Team robots are supposed to share a *cooperative attitude* which allows them to exchange information when required by other team members, and to take decisions towards avoiding possible conflicts through sharing resources and/or assuming goals (tasks). When the CC sends requests for helping impaired elderlies, the robot team first decides which member of the team will assume the task specified in the request, and then reallocates the current goals in order to satisfy all demands. The generic process to cooperatively decide who will assume the task is explained next and shown in Fig. 2. So, each mobile robot:

- generates a goal representing the task to be achieved,
- estimates the cost to achieve the goal specified in the request,
- sends its estimated cost to the team members,
- receives estimated costs from team members, and,
- takes a decision to assume the goal based on the estimations received from its peers.

Three cases might happen. (C1) The agent has the best estimation: it sends its peers the proposal to achieve the goal, and waits to receive their confirmation. (C2) There are other team mates better suited than itself to achieve the goal: it sends the agreement for them to achieve the goal. (C3) The agent has the optimal cost, but it is tied with other team mates: the tied peers add a randomly generated number to their estimations and send the new estimation to tied peers in order to allow one of them to accept the goal.

Goal allocation and cost estimation. A formal definition for the multi-robot goal allocation problem is as follows. Let $\{R_1, R_2, \ldots, R_w\}$ be a team of robots, which should achieve a number of goals, $G(N_1), G(N_2), \ldots, G(N_t)$, where a $G(N_p)$ goal consists on helping impaired elderly (needy) situated in specific locations. Goals are prioritized according to the level of assistance needed by the person. Let $Pri(N_n)$ be the priority to help the needy N_n. Then the priority of goal $G(N_n)$ that is helping the needy N_n will also be $Pri(N_n)$.

Let us suppose that at time t the robot R_r accepts an ordered set of prioritized goals called the *Robot Load*; $RL_{R_r}(t) = \{G_{R_r}(N_1(t_1)), G_{R_r}(N_2(t_2)), \ldots, G_{R_r}(N_{z-1}(t_{z-1})), G_{R_r}(N_z(t_z))\}$, where priority $Pri(G_{R_r}(N_p(t_p))) \geq Pri(G_{R_r}(N_k(t_k)))$, and $p < k$, $p, k = 1 \ldots z$. Notice that (1) t_t is the notification time of needy N_t, (2) $Pri(G_{R_r}(N_t(t_t)))$ is the priority of goal $G(N_t)$ included in $RL_{R_r}(t)$; (3) $N_t(t_t)$ makes reference to needy N_t, which was notified at time t_t, and (4) older people with equal priority are sorted according to a first-in/first-out notification time base, that is, the first elderly to be assisted would be the elderly with the earliest notification time.

On the one hand, initially or when the robot has no goals ($RL_{R_r}(t)) = \Phi$), the cost to achieve a new goal $G_{R_r}(N_i)$, is estimated with a *function* $F_{eval}(G_{R_r}(N_i), t) = f(Tr_{R_rN_i}, Th_{R_rN_i}, Wr_{R_rN_i}, Wr_{R_r}(t)) \in \mathbb{R}$, where $Tr_{R_rN_i}$ is the time needed for the robot to reach the needy N_i; $Th_{R_rN_i}$ is the time needed to help the needy N_i; $Wr_{R_rN_i}$ is the energy needed by R_r to help N_i, and $Wr_{R_r}(t)$ is the total energy available at time t.

$Tr_{R_rN_i}$ depends on the R_r trajectory to reach N_i, and the average R_r speed in the trajectory. The robot's trajectories are represented as a vector of navigation points $\{P_1, P_2, \ldots, P_n\}$, where each point P_j is characterized by a triple of Cartesian coordinates $(X_{P_j}, Y_{P_j}, Z_{P_j})$. P_1 is the robot position R_r, and P_n is N_i position. It was assumed that R_r has a uniform rectilinear movement between two consecutive points belonging to the trajectory, then *the time to complete the trajectory* can be calculated as:

$$Tr_{R_rN_i} = \sum_{j=1}^{n-1} dist_{P_j,P_{j+1}}/speed_{R_rN_i} t_{P_j,P_{j+1}}$$

$$= [(X_{P_j} - X_{P_{j+1}})^2 + (Y_{P_j} - Y_{P_{j+1}})^2$$

$$+ (Z_{P_j} - Z_{P_{j+1}})^2]^{0.5}$$

where $speed_{R_rN_i}$ is the speed of R_r while trying to reach N_i. $Th_{R_rN_i}$, the time to help N_i, is a predefined

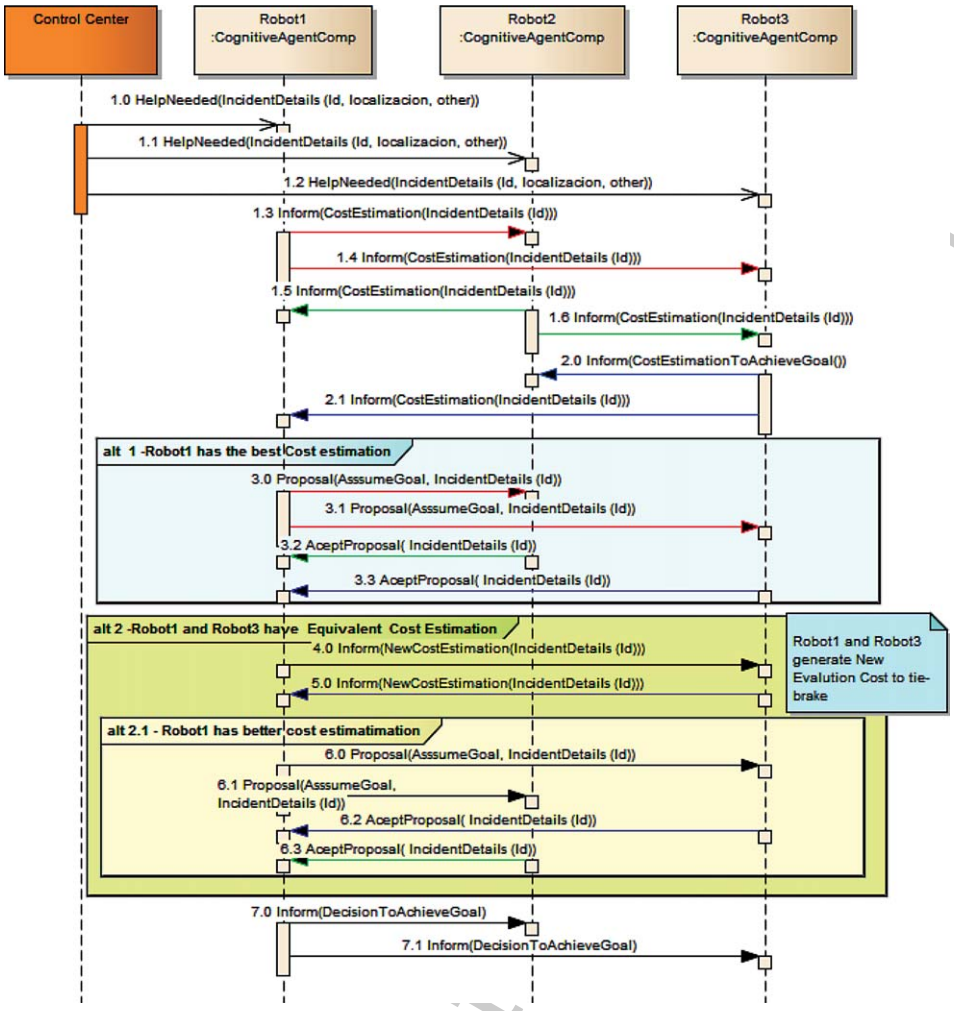

Fig. 2. The adaptive multi-agent system (AMAS) cooperation model.

constant proportional to N_i's priority. In practice it could be estimated by the CC or by the robot itself.

The robot gets its average energy consumption unit ($ACWr_{R_r}(t)$) and its remaining energy at time t ($Wr_{R_r}(t)$) from **the energy management component**. Then: (i) it estimates the energy needed to help N_i, $Wr_{R_rN_i} = (Tr_{R_rN_i} + Th_{R_rN_i}) * ACWr_{R_r}(t)$; (ii) it checks if there is enough energy to help N_i by comparing $Wr_{R_rN_i}$ with $Wr_{R_r}(t)$. If there is not enough energy the cost is estimated as (-1.0), otherwise as $Tr_{R_rN_i} + Th_{R_rN_i}$. That is,

$$\begin{aligned} &\textbf{if } (Wr_{R_r}(t) - Wr_{R_rN_i}) < 0 \\ &\textbf{then } F_{eval}(G_{R_r}(N_i), t) = (-1.0) \\ &\textbf{else } F_{eval}(G_{R_r}(N_i), t)) = (Tr_{R_rN_i} + Th_{R_rN_i}). \end{aligned}$$

On the other hand, when the robot has goals, $RL_{R_r}(t) = \{G_{R_r}(N_1), \ldots, G_{R_r}(N_k)\}$, $1 \leq k$, and the robot has enough energy to achieve all the goals in $RL_{R_r}(t)$ then

$$F_{eval}(RL_{R_r}, t) = \sum_{i=1}^{k} F_{eval}(G_{R_r}(N_i), t)$$

When the robot has a load $RL_{R_r}(t)$, the cost for a new goal $G(N_{new})$ is obtained by adding the new goal to RL_{R_r} and then evaluating the cost of

$$RL_{R_r}(t_{evalNnew}) = RL_{R_r}(t) \cup G_{R_r}(N_{new}).$$

As the priority of the new goal may involve reordering the current goals, and given that the location of N_{new} may change the current path to achieve all the goals in $RL_{R_r}(t_{evalNnew})$, the cost is estimated to verify

if there is enough energy to be achieved. The evaluation function used is:

$$\textbf{if } (Wr_{R_r}(t_{evalNnew}) - Wr_{R_r}(RL_{R_r}(t_{evalNnew}))) < 0$$
$$\textbf{then } F_{eval}(G_{R_r}(N_{new}), t_{evalNnew}) = (-1.0)$$
$$\textbf{else } F_{eval}(G_{R_r}(N_{new}), t_{evalNnew}) =$$
$$F_{eval}(RL_{R_r}, t_{evalNnew})$$

Team performance assessment. Performance evaluation of the goal allocation algorithm is based on the following three parameters. (1) The time required for a goal to be assigned to a team mate. This time is calculated using the processor real time clock as the time difference between the instant when the control center sends the request and the instant when the goal to help the elderly is accepted by a team mate. (2) Goal distribution among team members. (3) The cost of the robot team, which corresponds to the highest cost of the goals assumed by each team member.

Dealing with uncooperative peers. Cooperation comes out from the need of each agent to get information from its team mates to achieve their own goals. The cooperation process is highly dependent on team communication which quality cannot be guaranteed when the operating environment is under a critical situation. Cooperation might fail when communication is missing, and also due to internal processing factors such as lack of synchronization in the cooperation process, and malfunctioning of internal components like sensors, motion, vision, position, computing, and others. Consequently, each agent is able to deal with situations where: (i) they cannot communicate with their peers; (ii) communication is possible but team mates do not send the expected information, and/or they do not respond to requests; and, (iii) they send unexpected or outdated information. In these cases individual decisions should be taken to achieve the goals/tasks requested by the CC. To cope with "worst cases" which correspond to real situations the mobile eldercare robots team model has been extended to take into account the deadlines for decision making, missing information from the team mates, current robots' workload, and stressing requests from the CC.

A **hierarchical team model** has been implemented in order to have a reference for assessing the strengths and weaknesses of the AMAS model, and for the utilization of a "heavy deliberative control architecture" for implementing these models. The hierarchical team is made up of a **team leader** and a group of **subordinate robots** (see Fig. 3).

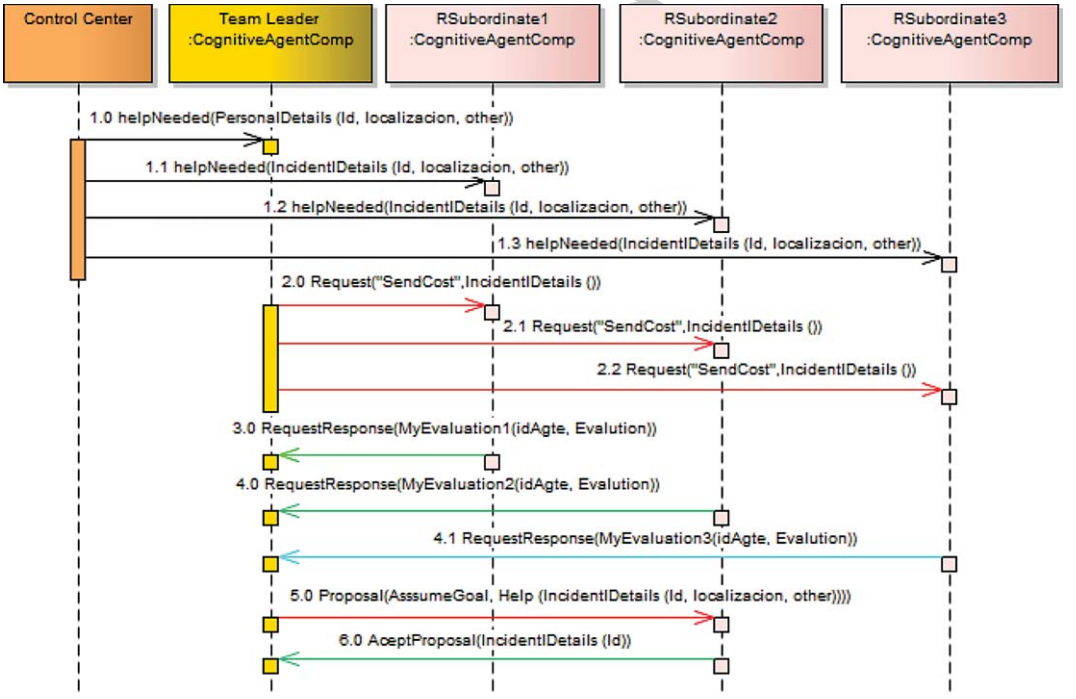

Fig. 3. The hierarchical cooperation model.

The team leader is in charge of interpreting the CC requests, and then assigning helping tasks to the best suited subordinate robot. The goal for each CC request is to decide which robot has to assume the task. To take this decision, the team leader solicits subordinates to send back their estimated cost to achieve the task. A proposal to achieve the task is sent to the subordinate with the best evaluation. The subordinate might reject the proposal explaining the reason, as for example internal troubleshooting issues, lack of energy to achieve all the tasks, or impossibility to access the target's location. Then, the team leader could either assign the task to another team member or ignore the reject and confirm its decision.

2.3. *Implementation approach using ICARO deliberative control pattern*

The ICARO framework has previously been used to model mobile robots with reactive patterns which control is based on finite state automata [15, 16]. The experimental setting implemented with ICARO is depicted in Fig. 4. The physical environment where the robots evolve is represented by a set of simulation components which manage the environment constraints, the human position and the robot movement. As the main focus of the work is on cooperative decision making, the robots are modeled with two concurrent components: the motion controller and the RGC component. The motion component is in charge of the interpretation and execution of movement control commands, and movement monitoring, which informs the RGC component about relevant motion states and control command execution. The RGC component is implemented with ICARO deliberative agent pattern which is based on a goal processor.

Robot behavior is characterized by: (i) the set of goals which can be achieved; (ii) the activities, processes and actions needed to achieve the goals; (iii) the information model representing the domain and environmental entities, the computing entities needed for representing goal achievement states, and intermediate results produced by activities and actions; and (iv) the process defining the life cycle of goals. This is performed through situation-action rules expressing conditions for (a) goal generation, (b) goal focalization, (c) goal achievement, and, (d) executing activities and actions to make it

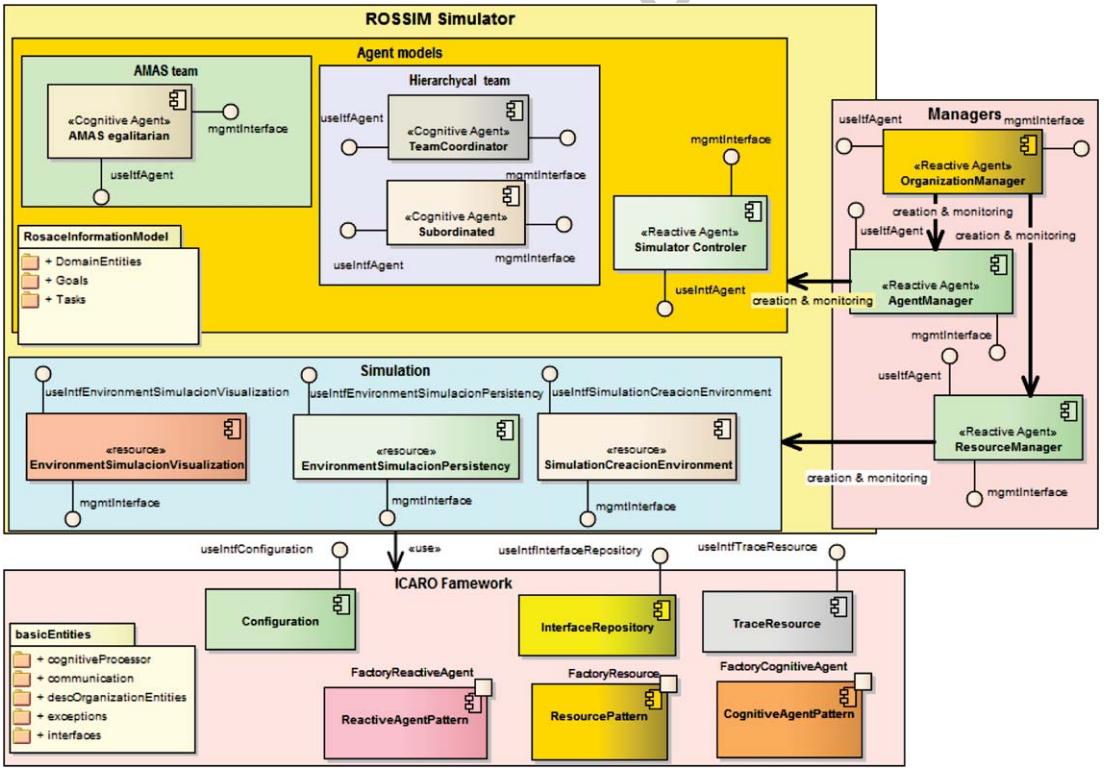

Fig. 4. Simulator architecture implemented with ICARO.

possible for pending goals to satisfy their achievement conditions.

Goals are represented as classes from which multiple object instances are generated. Activities and actions needed to achieve goals are represented as tasks. The work-flow of activities and actions needed to achieve goals are first defined with UML activity diagrams, and then implemented with situation-action rules. Also, multiple distributed deployment instances can be generated from each behavior model. The ICARO framework provides deployment, monitoring and communication transparency among component instances.

The **AMAS team model** is implemented with a common behavior model for all eldercare robots. Teams are made up of cloning instances; they have the same goals, tasks, information model, and goal-resolution rules. Requests sent by the CC are received by all team-members which generate similar goal instances: *helpNeedy()* and *decideWhoShouldGo()*. Cooperation is modeled in the protocol for making collective decisions, that is, to achieve the goal *decideWhoShouldGo()*. This is done by exchanging cost estimations, and then deciding which member of the team is the best situated to help the elderly. The goal resolution process is defined with 41 structured rules.

Although all team members voluntarily participate in the decision process, the way in which each robot achieves its own goals is dependent on its situation in the environment and on its internal state which is characterized by information objects in its working memory, including the previous goals and current focus representing the goal under resolution. Experimentation has been done for fine-tuning the model to allow the robot to take individual decisions when collective decisions fail, and to determine deadlines for expected information and for taking collective decisions. As most of these parameters are dependent of hardware and communication performance, they are defined as configurable.

The **hierarchical team model** has two roles implemented with two behavioral models. The *team leader* is in charge of interpreting the requests from the CC and deciding which team-member should be assigned to achieve the goal. The *subordinate robot* receives messages from its leader, first requesting to estimate its cost for achieving the goal, and then to accept/refuse proposals for assuming the goal. Subordinates might refuse proposals when they do not have the necessary means to achieve them. However, the final decision to assign the goal corresponds to the leader. Deadlines for expected answers and deadlines for taking decisions are similar to the AMAS model. The information model is the same as for AMAS, goal and tasks are also shared, but the leader role is implemented with 15 rules and the subordinate role with 6 rules.

The system is implemented in Java. It may run in a central node or component instances can be deployed in a network of processing nodes with Windows/Linux OS and virtual machine Java 6.xx. The rule processor used for implementing the deliberative agent pattern is based on Drools 5.x. [2] and communication among mobile robots is performed through RMI [19].

3. Experimental results

Metrics to assess both the model and the implementation approach using the deliberative architecture considers two main aspects: functional conformity and performance. Functional conformity focuses on the quality of goal allocation and goal distribution among team members. Performance considers the time needed for the team to assume goals for helping the needy requested by the CC. The decision making process is done while robots stop moving to help potential needy or due to obstacle detection. Information about the robot motion state is taken into account for participating in the collective decision making process and for changing motion directives when more priority goals are assumed by the robot. Then the motion component is required to calculate a new trajectory and move out to the position of the potential needy. Metric values have been gathered from testing experiments after considering the following parameters: (i) the team size and the number of elderlies to assist; (ii) the frequency of messages sent by the CC to assess the response of the team faced up to stressing requests; (iii) the deployment in different processing nodes to assess the impact of real parallel processing and communication.

Experimentation in one central node has been performed in an AMD Phenom II X4 processor at 3.20 GH with 4MB Ram and Windows 7 OS. The two additional nodes for distributed experiments are based on Intel core I7 at 2.20 Ghz with 8Gb of Ram, Windows 7 OS, and AMD Turion X2 at 2 Ghz, 2Gb of Ram and Windows XP OS. The most significant results are summarized below.

Natural, non-stressing requests. The **AMAS model** works as expected in situations where the CC sends requests at a frequency greater than the time needed for deciding the responsibility to assume the goal. As the time required to take decisions increases with the size of the team, deadlines for waiting responses and

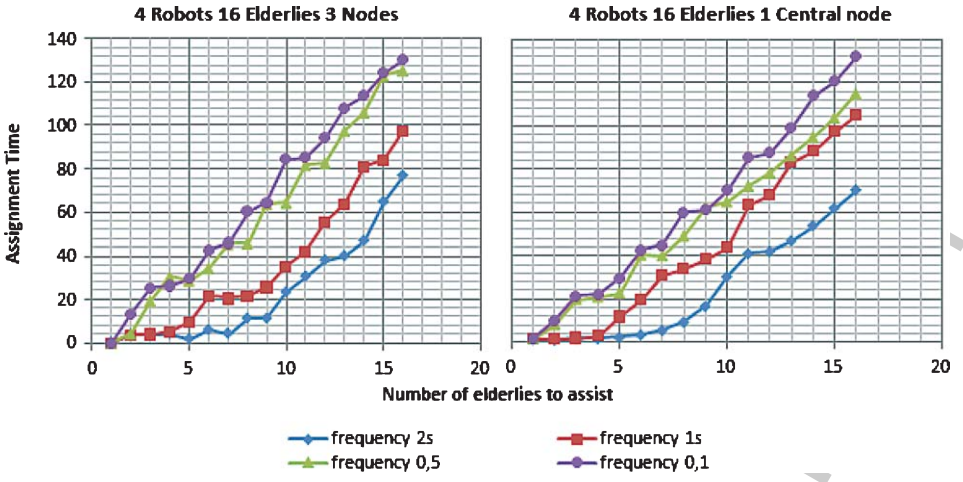

Fig. 5. AMAS model goal assignment.

taking decisions are also increased to synchronize goal resolution. When deadlines are not met, the same goal can be assumed by two or more team members, but this rarely happens. Tie-brakes for cost evaluation are satisfactorily solved. Fig. 5 shows performance results for mobile eldercare robots deployed in one central node and deployed in 3 nodes. Time for allocating goals is quite similar.

Stressing requests. High frequency requests degrade team performance due to the perturbation caused by the interpretation of incoming requests during collective decision making. The first consequence of increasing the frequency of CC requests is desynchronizing the process for achieving goals. CC messages are received at different time and processed at different speed by team-peers. When a team-member receives a request from the CC, it retrieves the elderly's priority and generates new goals for helping the needy and for deciding which robot should assume that goal. If the priority of the new needy is higher than the senior whose decision is trying to achieve, it delays the resolution of the current goal and starts a new decision process to help this new elderly. It is assumed that its team mates will do the same; consequently it estimates its cost to achieve the goal and sends it to its companions.

Task assumption through team collaboration. It may occur that team-peers receive cost estimations and requests for sending their estimations before the message from the CC is processed. This lack of synchronization might lead various peers to take the responsibility to assume the same goal. To deal with this situation, the peer receiving cost estimations, or requests for sending estimations about unknown elderlies, acts as if it were informed by its peer about the CC request. It trusts peer's information, and then it generates the goals and starts participating in the decision process. When the CC request arrives, the interpretation is already done. If the CC request cannot be received, the robot is indirectly informed by its team mates.

Hierarchy versus AMAS. Goal desynchronizing in the AMAS model delays decisions due to multiple interruptions during the decision process, and, consequently, decreases team performance, but the goals are still correctly allocated. Experimentation shows a progressive degradation of performance when stressing demand increases, although quality is still assured (see Fig. 6). This confirms the robustness of the model. Centralization of CC message interpretation and decision making facilitates conflict resolution, reducing the number of messages needed for goal assignment.

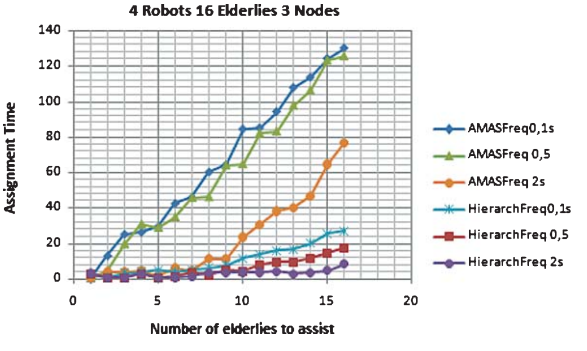

Fig. 6. AMAS versus hierarchical model.

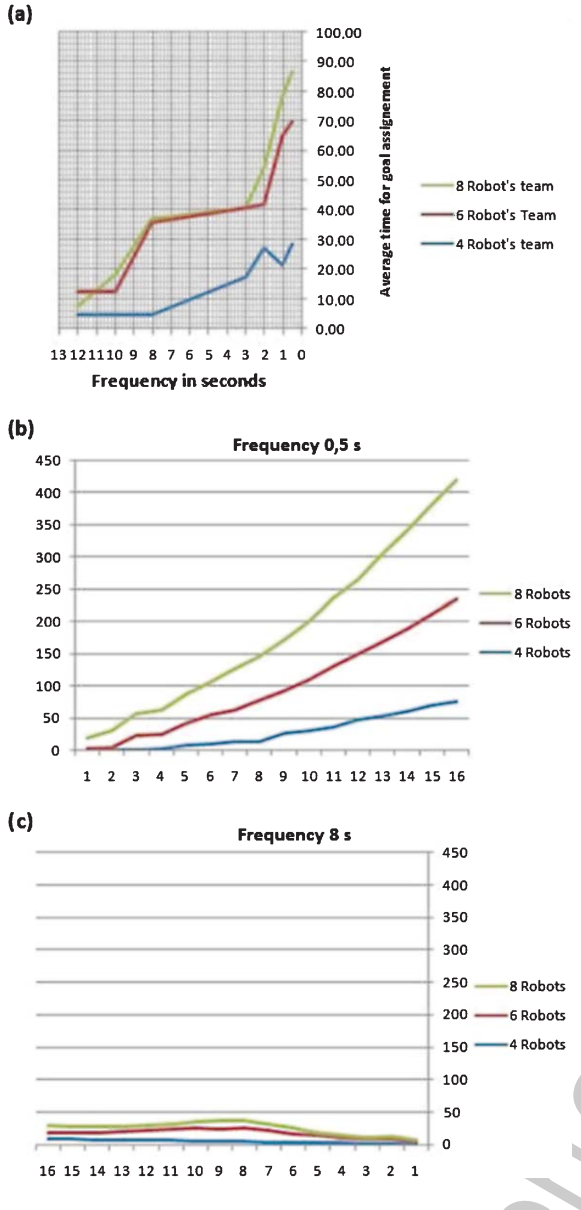

Fig. 7. Performance of AMAS model in experiments with different team size.

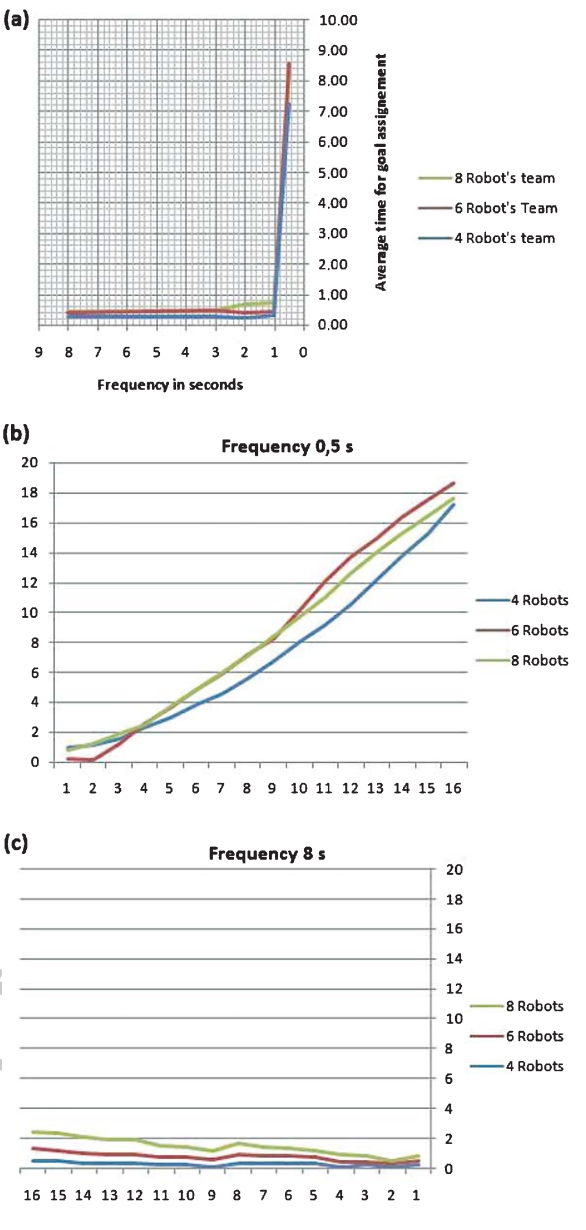

Fig. 8. Performance of hierarchical model in experiments with different team size.

Performance with respect to the AMAS model is also shown in Fig. 6. The hierarchical model is 10 times faster than the AMAS model. Nonetheless, stress has more impact on its performance. Stressing requests degrade performance by a factor of 3.3 while the impact in AMAS is 1.6. The main weakness of this model concerns robustness since the efficiency of the team is dependent on the decisions of the leader. The team becomes inactive when the leader or the communication among the leader and the subordinates fails.

Moreover, other experiments in a central node have been performed in an Intel core I7 at 2.20 GHz with 16 GB of RAM, Windows 7 OS (see Figs. 7 and 8). This experimentation has been carried out for the purpose of evaluating the performance of AMAS and hierarchical models when changing the number of robots (4, 6 and 8 robots) and the frequency of messages sent by the CC (0.5, 1, 2, 3, and 8 seconds). The number of elderlies being assisted is fixed to 16. In this case, the

metric average time for goal assignment (assigning help to one elderly) is used as comparison parameter. The most significant results are summarized as follows:

- The average time for goal assignment increases when the team size increases (see Figs. 7a and 8a). The degradation of performance also increases.
- The average time for goal assignment decreases when the frequency of messages sent by the CC decreases; that is, messages are sent using a higher time interval (see Figs. 7a and 8a).
- The average time for goal assignment converges faster in the hierarchical model (see Fig. 8a) than in the AMAS model (see Fig. 7a). For example, in the hierarchical model the average time obtained does not change when the frequency is greater than or equal to 1 second (see Fig. 7a).
- The trends and patterns in the obtained data are more variable when the elapsed time to send a new request by the control center decreases (e.g., frequency 0.5 versus 8 seconds; see Figs. 7b and 7c for AMAS model; and Figs. 8b and 8c for hierarchical model.

In short, according to the metric average time for goal assignment, the new experiments confirm that the hierarchical model is faster than the AMAS model, but it is still dependent on the resilience of the coordinator.

4. Conclusions and future challenges

Experimentation with decision models using deliberative architectures requires the availability of engineering tools which facilitate quick development, deployment and evaluation. Despite the wide number of papers devoted to team modeling, availability of systems allowing verification and extension of these models are scarce. This work has faced two related challenges, namely, model validation taking into account realistic constraints, and engineering evaluation mainly focused on the utilization of heavy deliberative architectures for controlling the behavior of complex entities such as mobile eldercare robots.

Experimentation has gone beyond *best cases* to be focused on stressing test cases to validate key aspects of cooperative decision making such as performance, quality and robustness. The most significant results are obtained in worse case scenarios where team members face up with internal and communication failure, and stressing requests. AMAS performance is significantly lower than the hierarchical model one. However, this weakness might be compensated by higher robustness. Stress decreases performance in both models, most significantly in the hierarchical model, but quality is guaranteed. The utilization of an encapsulated deliberative architecture facilitates high level modeling, and the traceability of the collaborative decision making process, then allowing incremental development and bridging the gap between analysis, design and implementation. Seemly creation of multiple parallel instances is done without penalizing deployment and performance.

The current system is made up of open source re-usable components provided by the ICARO framework. Extensibility, manageability, integration and deployment can be done with most popular Integrated Development Environments (IDE). This paves the way to the development and experimentation with new team models where team mates change their role dynamically. For example, the implementation of a team which starts hierarchical but becomes AMAS when the coordinator (leader) loses connection with its peers can be performed without significant effort. Other models such as selecting a new leader or creating a partial hierarchy for big teams might be quickly developed.

The current version of the simulator facilitates experimentation with different team size and person location, however it should be extended to deal with dynamic robot failure and creation and execution of more complex scenarios. The next step is to go beyond simulation to validate the models incorporated into current mobile eldercare robots navigating in a physical elderly care environment.

Acknowledgements

This work is supported by the French RTRA-STAE foundation (Réseau Thématique de Recherche Avancée, Sciences et Technologies pour l'Aéronautique et l'Espace) in the scope of the ROS-ACE project. This work is also supported by the Spanish Ministerio de Economía y Competitividad/FEDER under project TIN2010-20845-C03-01.

References

[1] J.C. Augusto, H. Nakashima and H. Aghajan, Ambient intelligence and smart environments: A state of the art, *Handbook of Ambient Intelligence and Smart Environments* (2010), 3–31.

[2] M. Bali, Drools JBoss Rules 5.0 Developer's Guide, *Packt Publishing* (2009).
[3] P. Benjamin, D. Lyons and D. Lonsdale, ADAPT: A cognitive architecture for robotics, *Proceedings of the International Conference on Cognitive Modelling*, 2004, pp. 337–338.
[4] Z.Z. Bien and H.E. Lee, Effective learning system techniques for human-robot interaction in service environment, *Knowledge-Based Systems* **20** (2007), 439–456.
[5] D. Brugali and P. Scandurra, Component-based robotic engineering (Part I), *IEEE Robotics & Automation Magazine* **16** (2009), 84–96.
[6] C. Burghart, R. Mikut, R. Stiefelhagen, T. Asfour, H. Holzapfel, P. Steinhaus and R. Dillmann, A cognitive architecture for a humanoid robot: A first approach, *Proceedings of the 5th International Conference on Humanoid Robots*, 2005, pp. 357–362.
[7] M. Cirillo, L. Karlsson and A. Saffiotti, Human-aware planning for robots embedded in ambient ecologies, *Pervasive and Mobile Computing* **8** (2012), 542–561.
[8] J. Chakraborty, A. Konar, L.C. Jain and U.K. Chakraborty, Cooperative multi-robot path planning using differential evolution, *Journal of Fuzzy and Intelligent Systems* **20** (2009), 13–27.
[9] Â. Costa, J.C. Castillo, P. Novais, A. Fernández-Caballero and R. Simoes, Sensor-driven agenda for intelligent home care of the elderly, *Expert Systems with Applications* **39** (2012), 12192–12204.
[10] M.B. Dias, R. Zlot, N. Kalra and A. Stentz, Marketbased multirobot coordination: A survey and analysis, *Proceedings of the IEEE* **94** (2006), 1257–1270.
[11] A.C. Domínguez-Brito, F.J. Santana-Jorge, J. Cabrera-Gámez, J.D. Hernández Sosa, J. Isern González and E. Fernández- Perdomo, Exploring interfaces in a distributed component-based programming framework for robotics, *Proceedings of the 4th International Conference on Agents and Artificial Intelligence*, 2012, pp. 667–672.
[12] G. Echeverria, N. Lassabe, A. Degroote and S. Lemaignan, Modular open robots simulation engine: MORSE, *Proceedings of the IEEE International Conference on Robotics and Automation*, 2011, pp. 46–51.
[13] S. Franklin, A. Finn, J. Pattison and L.C. Jain, Towards scene understanding using a co-operative of robots, *Journal of Fuzzy and Intelligent Systems* **21** (2010), 101–112.
[14] F. Garijo, S. Bravo, J. Gonzalez and E. Bobadilla, BOGAR LN: An agent based component framework for developing multi-modal services using natural language, *Lecture Notes in Artificial Intelligence* **3040** (2004), 207–220.
[15] J.M. Gascueña, A. Fernández-Caballero and F.J. Garijo, Using ICARO-T framework for reactive agent-based mobile robots, *Advances in Soft Computing* **70** (2010), 91–101.
[16] J.M. Gascueña, E. Navarro and A. Fernández-Caballero, VigilAgent for the development of agent-based multi-robot surveillance systems, *Proceedings of the 5th KES International Conference Agent and Multi-Agent Systems: Technologies and Applications*, 2011, pp. 200–210.
[17] J.-P. Georgé, M.-P. Gleizes, , F.J. Garijo, V. Noël and J.P. Arcangeli, Self-adaptive coordination for robot teams accomplishing critical activities, *Advances in Intelligent and Soft Computing* **70** (2010), 145–150.
[18] J.P. Georgé, M.-P. Gleizes and V. Camps, Cooperation, *Self-organising Software: From Natural to Artificial Adaptation*, 2011, pp. 193–226.
[19] W. Grosso, Java RMI, *O'Reilly Media* (2001).
[20] C. Jang, S. Lee, S. Jung, B. Song, R. Kim, S. Kim and C. Lee, OPRoS: A new component-based robot software platform, *ETRI Journal* **32** (2010), 646–656.
[21] J. Lacoûture, V. Noël, J.P. Arcangeli and M.-P. Gleizes, Engineering gent frameworks: An application in multi-robot systems, *Advances in Intelligent and Soft Computing* **88** (2011), 79–85.
[22] J. Lacoûture, J.M. Gascueña, M.-P. Gleizes, P. Glize, F.J. Garijo and A. Fernández-Caballero, ROSACE: Agent-based systems for dynamic task allocation in crisis management, *Advances in Soft Computing* **155** (2012), 255–259.
[23] A. Mordenti, Programming Robots with an Agent-Oriented BDI-based Control Architecture: Explorations using the JaCa and WeBots platforms, Doctoral Thesis, Alma Mater Studiorum Universitá di Bologna, (2012).
[24] D. Nakhaeinia and B. Karasfi, A behavior-based approach for collision avoidance of mobile robots in unknown and dynamic environments, *Journal of Fuzzy and Intelligent Systems* **24** (2013), 299–311.
[25] H.S. Park and S.B. Cho, A modular design of Bayesian networks using expert knowledge: Context-aware home service robot, *Expert Systems with Applications* **39** (2012), 2629–2642.
[26] A. Sanfeliu, N. Hagita and A. Saffiotti, Network robot systems, *Robotics and Autonomous Systems* **56** (2008), 793–797.
[27] S. Schiffer, A. Ferrein and G. Lakemeyer, CAESAR: An intelligent domestic service robot, *Intelligent Service Robotics* **5** (2012), 259–273.
[28] M.V. Sokolova, J. Serrano-Cuerda, J.C. Castillo and A. Fernández-Caballero, A fuzzy model for human fall detection in infrared video, *Journal of Fuzzy and Intelligent Systems* **24** (2013), 215–228.
[29] S. Suárez, C. Quintero and J.L. de la Rosa, A real time approach for task allocation in a disaster scenario, *Proceedings of the 8th International Conference on Practical Applications of Agents and Multiagent Systems*, 2010, pp. 157–162.
[30] C. Wei and K.V. Hindriks, An agent-based cognitive robot architecture, *Lecture Notes in Artificial Intelligence* **7837** (2013), 54–71.